Probit latent variables estimation for a gaussian process classifier: Application to the detection of high-voltage spindles

Rémi Souriau, Vincent Vigneron, Jean Lerbet, Hsin-Chen Chen

Probit latent variables estimation for a Gaussian Process classifier. Application to the detection of High-Voltage Spindles

Rémi Souriau*, Vincent Vigneron, Jean Lerbet, and Hsin Chen

IBISC EA 4526, Université d'Evry, Université Paris Saclay, France
National Tsing Hua University, Electrical engineering department, Hsinchu, Taiwan
{remi.souriau,vincent.vigneron,jean.lerbet}@ibisc.univ-evry.fr
hchen@ee.nthu.edu.tw

Abstract. The Deep Brain Stimulation (DBS) is a surgical procedure efficient to relieve symptoms of some neurodegenerative disease like the Parkinson's disease (PD). However, apply permanently the deep brain stimulation due to the lack of possible control lead to several side effects. Recent studies shown the detection of High-Voltage Spindles (HVS) in local field potentials is an interesting way to predict the arrival of symptoms in PD people. The complexity of signals and the short time lag between the apparition of HVS and the arrival of symptoms make it necessary to have a fast and robust model to classify the presence of HVS ($Y = 1$) or not ($Y = -1$) and to apply the DBS only when needed. In this paper, we focus on a Gaussian process model. It consists to estimate the latent variable f of the probit model: $\Pr(Y = 1|input) = \Phi(f(input))$ with Φ the distribution function of the standard normal distribution.

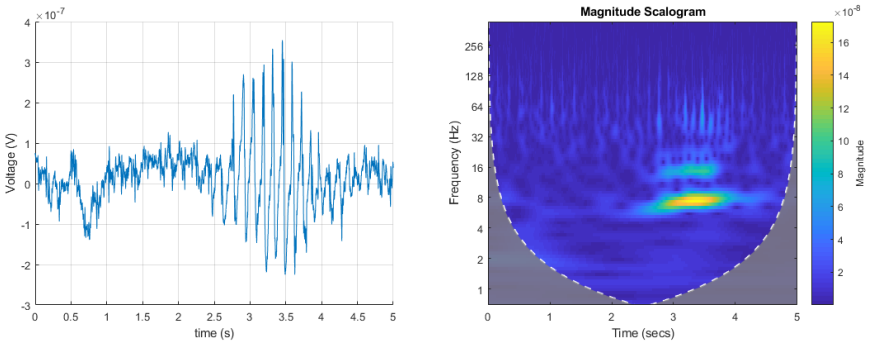
Keywords: Deep Learning, Gaussian Processes, Autoencoder, Classification, High-Voltage Spindle, Parkinson Diseases

1 Introduction

The Parkinson's disease (PD) is a progressive *neurodegenerative* disease. The depletion of the dopamine in the basal ganglia network leads to several symptoms like rigidity, posture instability, slow motion or pain for example. The expectation of the number of PD victims in Asian countries is 6.17 millions in 2030 [2]. The deep brain stimulation (DBS) is a surgical procedure used to relieve disabling neurological symptoms for diseases like PD [8]. A high-frequency stimulation signal (around 130Hz) is continuously applied to a deep-brain region called the *subthalamic nucleus* (STN) to relieve the symptoms. The main drawback of the DBS is the absence of any control on stimulation to minimize side effects. In addition, contemporary DBS implant requires another surgery to

* This work was partly supported by the National Tsing Hua University (Hsinchu, Taiwan) and Ministry of Science and Technology, R.O.C. (Taiwan).

replace battery every 6 or 7 years.
Recent studies show we can predict the arrival of PD symptoms by the detection of high-voltages spindles (HVS) in recorded signals in local field potentials (LFPs) [1]. The HVS signals as *e.g.* in Fig. 1 are synchronous spike-and-wave patterns in LFPs oscillating in the 5-13 Hz frequency band. Suppressing HVS signals is found useful for delaying the progress of PD and deleting symptoms. Being able to detect HVS make possible the realization of a *closed-loop system* to control the DBS. However the diffusion of signals in the brain is *nonlinear* and there is only few milliseconds between the HVS wave and the apparition of PD symptoms. Hence a fast and robust model is needed for real time HVS detection and to apply the high frequency signal only when it is needed.



(a) Temporal representation.

(b) Cosine wavelet transform using Morse wavelet.

Fig. 1: Signals recorded in LFPs in two different representations. HVS are located between 2.5 and 5 seconds. HVS are characterized by a fundamental frequency between 5 and 13 Hz.

In this paper, the PD rat model is used. Data are collected from eight intracortical channels from different cortical regions. In this paper, we investigate performance of the Gaussian Process (GP) [3] for the detection of HVS. The GP model is a *Bayesian network* with continuous variables. Bayesian network model relations of *causality* between variables and in our study, data collected are the result of a diffusion of signals between neurons in the brain. Moreover, relations between variables model by a GP are nonlinear. Section 2 presents how data are collected and the preprocessing of data. Details of the model are developed in section 3. The two last sections give main results and discuss some future improvement and other possible approaches.

2 Data collection

2.1 Data Acquisition

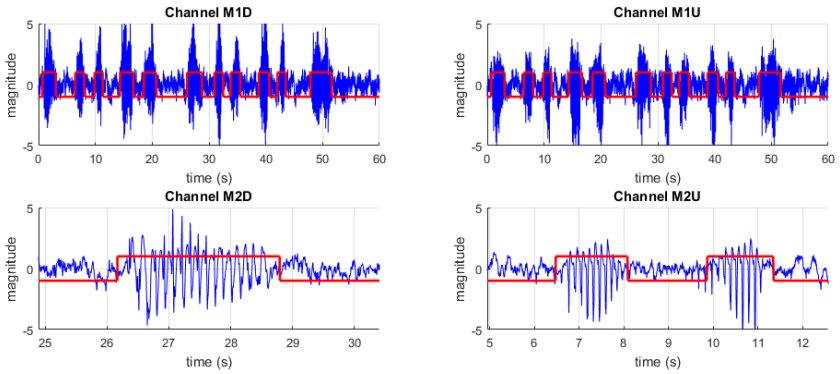
The PD rat model has been used to develop and evaluate the results. The description of the procedure to extract data is given in [7, sect. 2]. The LFPs were recorded from eight different brain regions listed in Tab.1. The frequency sampling of signals was 1 kHz and the recording duration of one session was 60 seconds (60,000 samples). Several sessions have been recorded on PD rats.

Notation	**Region name**
M1D	Layer 5b of the primary motor cortex
M1U	Layer 2/3 of the primary motor cortex
M2D	Layer 5b of the secondary motor cortex
M2U	Layer 2/3 of the secondary motor cortex
SD	Layer 5b of the primary somatosensory cortex
SU	Layer 2/3 of the primary somatosensory cortex
STRI	Dorsal region of striatum
THAL	Ventrolateral thalamus

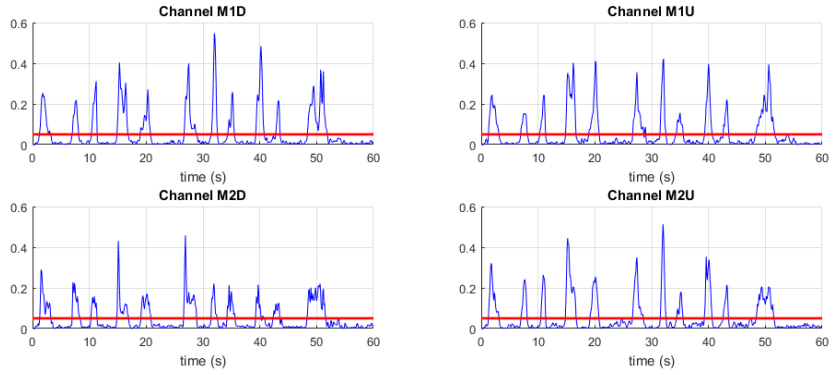
Table 1: List of brain region where LFPs signals were recorded.

2.2 Data preparation

GP classifier is a model which requires a *supervised* learning. The data preparation step consists to construct *feature* signals from collected data to *train* the model and feature signals for *testing* the model. One 60s session has been used for the training step and the other session has been used for the testing step. The preprocessing step is the same for the two data sets. HVS wave's fundamental frequency is between 5 and 13 Hz and, according to observations in Fig. 1b, harmonics of HVSs are visible around 30 Hz. High frequencies noises and continuous components of signals were deleted with a second order Butterworth filter between 1 and 200 Hz while preserving much of the HVS frequency content. Each channel was normalized independently from each other (zero mean and unit variance).
Then we define the prediction class vector $Y_n \forall n$ for the training set and the testing set. The presence of HVS is characterized by the apparition of *spike-and-wave* patterns with a fundamental frequency between 5 and 13 Hz. We estimate the Power Spectral Density (PSD) with a periodogram using the Hanning window with a 500 ms time window each 100 ms. Then by computing the PSD mean between 5 and 13 Hz and performing an interpolation, we plot the PSD mean between 5-13 Hz as a function of time (see Fig. 2). We then defined our *ground truth* by thresholding our observations above at least one quarter of the signal magnitude. Result are given in Fig. 2.



(a) PSD mean as a function of time for channels M1D, M2D, M1U and M2U. The red line represent the ground truth: if $\frac{3}{4}$ of signal magnitude is above the threshold, then we consider we detect the presence of the HVS.



(b) Result of the classification on the four same channels. Channels MD2 and MU2 are zoomed to observe with more precision the switch of state HVS/no HVS and reverse.

Fig. 2: data preparation step.

3 Model

3.1 Gaussian process classifier

A closed-loop DBS system delivers the stimulation only when needed.
Mathematically this consists to build a two-class classifier $\mathcal{C}$ capable to identify the presence or the absence of HVS. The available information for the classification are the values of the R channels and p previous signal values for each channel. Let $X_n \in \mathbb{R}^{R\times(p+1)}$ denotes the concatenated feature vector recorded between times n and $n-p$ and $Y_n \in \{-1, 1\}$ be the associated output of the *supervised classifier*. Suppose also the database is shared in a training set for which Y_n is known and a test set for which Y_n is *unknown*. The aim of the model is to estimate Y_n from new observations X_n.
We note in the following $\mathcal{D} = \{X_n, Y_n\}_{n\in[1,N]}$ the training set with $X = \{X_n\}_{n\in[1,N]}$ being independent randomly selected input observations and $Y = \{Y_n\}_{n\in[1,N]}$ the associated output decision respectively. The GP classifier focus on modeling the *posterior* probabilities by defining the *latent* variables $f_n = f(X_n)$.
The model used here is the *probit* model: $\Pr(Y = 1|X) = \Phi(f(X))$ where Φ denotes the cumulative density function of the standard normal distribution. The likelihood of the probit model with independent observations and given $f = \{f(X(n))\}_{n\in[1,N]}$ is:

$$p(Y|f) = \prod_{n=1}^{N} p(Y_n|f_n) = \prod_{n=1}^{N} \Phi(Y_n f_n). \tag{1}$$

In a GP, f is a stochastic process which associates a zero mean normal random value for an input $X(n)$. For the training set $\mathcal{D}$ we have $p(f|X, \Theta) \sim \mathcal{N}(0, \mathbf{C}_N)$ where Θ is a set of hyper-parameters and $\mathbf{C}_N$ is a covariance matrix modelized with a *squared exponential* and a Gaussian noise [6]:

$$\mathbf{C}_N(X_i, X_j) = \theta_0^2 \exp\left(-\frac{1}{2}\sum_{n=1}^{dim(X_i)} \frac{(X_i^{(n)} - X_j^{(n)})^2}{\lambda_n^2}\right) + \theta_1^2 \delta_{(X_i,X_j)}. \tag{2}$$

$X_i^{(n)}$ is the nth component of X_i and $\delta_{(\cdot)}$ is the Kronecker delta. The set of hyper-parameters Θ is composed of $\left\{\theta_1, \theta_2, \{\lambda_n\}_{n\in[1,N]}\right\}$. Baye's posterior probability rule of the latent variable f with Θ known can be written:

$$p(f|\mathcal{D}, \Theta) = \frac{p(Y|f)p(f|X, \Theta)}{p(\mathcal{D}|\Theta)} = \frac{\mathcal{N}(f|0, \mathbf{C}_N)}{p(\mathcal{D}|\Theta)} \prod_{n=1}^{N} \Phi(Y_n f_n). \tag{3}$$

With the marginalization of Eq. (3) for a new observation X_{N+1} we obtain:

$$\Pr(f_{N+1}|\mathcal{D}, \Theta, X_{N+1}) = \int \Pr(f_{N+1}|f, X, \Theta, X_{N+1}) \Pr(f|\mathcal{D}, \Theta) \mathrm{d}f, \tag{4}$$

and the expectation of the Eq.4 gives:

$$\Pr(Y_{N+1}|\mathcal{D},\Theta,X_{N+1}) = \int \Pr(Y_{N+1}|f_{N+1})\Pr(f_{N+1}|\mathcal{D},\Theta,X_{N+1})\mathrm{d}f_{N+1} \tag{5}$$

We model the posterior probability $q(f|\mathcal{D},\Theta) \sim \mathcal{N}(m,A)$ to compute $\Pr(Y_{N+1} = 1|\mathcal{D},\Theta,X_{N+1})$. And then, for a new observation $N+1$, we can show that the posterior probability of f_{N+1} is $q(f_{N+1}|\mathcal{D},\Theta,X_{N+1}) \sim \mathcal{N}(\mu,\sigma)$ with:

$$\begin{cases} \mu &= k^T\mathbf{C}_N^{-1}m, \\ \sigma^2 &= \kappa - k^T(\mathbf{C}_N^{-1} - \mathbf{C}_N^{-1}A\mathbf{C}_N^{-1})k. \end{cases} \tag{6}$$

where $k = (\mathbf{C}_N(X_1,X_{N+1}),\ldots,\mathbf{C}_N(X_N,X_{N+1}))^T$ is the covariance function vector between each observation of the training set and the new observation X_{N+1} and $\kappa = \mathbf{C}_N(X_{N+1},X_{N+1}) = \theta_0^2 + \theta_1^2$ is the variance of X_{N+1}.
With the approximation of $\Pr(f|\mathcal{D},\Theta)$, Eq. (5) becomes:

$$\Pr(Y_{N+1} = 1|D,\Theta,X_{N+1}) = \Phi\left(\frac{\mu}{\sqrt{1+\sigma^2}}\right) \tag{7}$$

Training a GP consist to find Θ, m and A. We can learn Θ by computing the log-likelihood of the log posterior probability $\log q(Y|X,\Theta)$ Eq.8 (see [6, Chapter 5]) and his gradient in function of Θ.

$$\log q(Y|X,\Theta) = -\frac{1}{2}f^T\mathbf{C}_N^{-1}f + \log p(Y|f) - \frac{1}{2}\log\det\left(I + W^{\frac{1}{2}}\mathbf{C}_N W^{\frac{1}{2}}\right) \tag{8}$$

With $W = -\Delta_f \log p(Y|f)$ (Hessian) and f such the unormalized log likelihood $\log p(f|\mathcal{D},\Theta)$ is maximized:

$$\begin{aligned} \log p(f|\mathcal{D},\Theta) &= \log p(Y|f) + \log p(f|X) \\ &= \log p(Y|f) - \frac{1}{2}f^T\mathbf{C}_N^{-1}f - \frac{1}{2}\log\det(\mathbf{C}_N) - \frac{N}{2}\log 2\pi. \end{aligned} \tag{9}$$

For a given Θ, we can find $m = \arg\max_f \log p(f|\mathcal{D},\Theta)$ by using the Newton's method. Eq.10 and Eq.11 give the gradient and the hessian of $\log p(f|\mathcal{D},\Theta)$, respectively.

$$\nabla_f \log p(f|\mathcal{D},\Theta) = \nabla_f \log p(Y|f) - \mathbf{C}_N^{-1}f. \tag{10}$$

$$\Delta_f \log p(f|\mathcal{D},\Theta) = \Delta_f \log p(Y|f) - \mathbf{C}_N^{-1}. \tag{11}$$

The maximization of $\log p(f|\mathcal{D},\Theta)$ makes use of the first and second order partial derivation of $\log p(Y|F)$ in function of f_i.

$$\frac{\partial}{\partial f_i}\log p(Y|f) = \frac{Y_i\phi(f_i)}{\Phi(Y_i f_i)}. \tag{12}$$

$$\frac{\partial^2}{\partial f_i^2}\log p(Y|f) = -\frac{\phi(f_i)^2}{\Phi(Y_i f_i)^2} - \frac{Y_i f_i \phi(f_i)}{\Phi(Y_i f_i)}. \tag{13}$$

Where $\phi(.)$ is the density function of the standard normal distribution. Learning m allow to compute Eq.8 and their gradient in function of Θ. We implement a gradient descent search of the optimum Θ^* that leads to the following iterative algorithm:

$$\Theta^{(k+1)} = \Theta^{(k)} - \alpha_k \nabla_\Theta \log q(Y|X,\Theta)). \tag{14}$$

But Eq. (14) requires to inverse the $N \times N$ matrix $\mathbf{C}_N$ at each iteration which can be time consuming for a large number of observations. Once m and Θ found, we can compute $A = \left(\mathbf{C}_N^{-1} + W\right)^{-1}$.
Finally, the GP classifier is learned by identifying the covariance matrix between observations $\mathbf{C}_N$ as a function of the hyper-parameters Θ, the mean vector m is learning for each iteration of Θ then and the covariance matrix A is deduced.
Once the learning is done, the prediction step consists to compute the covariance vector k between the new observation X_{N+1} and the training set X and then estimates the probability $\Pr(Y_{N+1} = 1|\mathcal{D}, \Theta, X_{N+1})$. If $\Pr(Y_{N+1} = 1|\mathcal{D}, \Theta, X_{N+1}) > 0.5$ then $Y_{N+1} = 1$ and $Y_{N+1} = -1$ else.

3.2 Input *autoencoding*

Learning the model consist in two step: learning the hyper-parameters Θ and learning the parameters of $q(f_{N+1}|\mathcal{D}, \Theta, X_{N+1})$. HVS have a fundamental frequency between 5 and 13 Hz. with a the maximal period of 200 ms. Choosing $p = 199$ to have at least one period of the signal leads to a model with high dimensions: the input size of X_n is then $(p+1) \times R = 1600$ and Θ has 1602 parameters. To reduce the the dimensionality of the input vector X_n (which makes it difficult to use for real time applications) we use an autoencoder (see Fig. 3). This autoencoder consists in a 3 layers neural network that compresses input data onto the hidden layer. We present to the input and the output layers the same input vector X_n. The activation function of the hidden layer is sigmoïdal function $s(\cdot)$ that permits nonlinear combination of the inputs:

$$s(x_j) = \frac{1}{1 + \exp(-b_j - \sum_i w_{ij} x_i)} \tag{15}$$

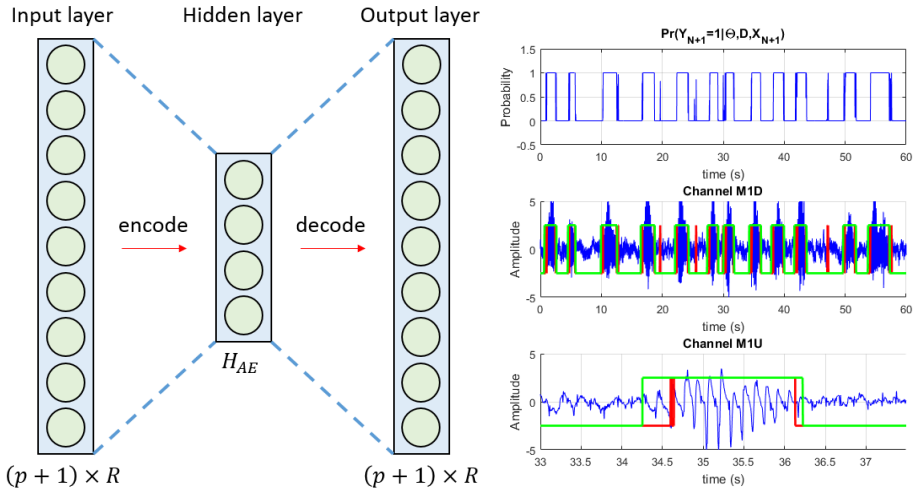
where $(x_1, \ldots, x_p)^T$ is the input vector. Learning this autoencoder consists to find biases b_j and weights w_{ij} of the input neurons. The output layer has to be the closest possible to the input layer. The training algorithm is the *scaled conjugate gradient* [4] using the mean square error with L_2 sparsity regularized loss function [5].

Fig.3 gives an example of result for a number of observations $N = 500$ and the size of the autoencoder $H = 10$. The sensitivity and the specificity (see section 4) are, respectively, 82.92% and 99.31%.

4 Experimental results

Detection of HVS has been applied on different rats with various set of parameters for the learning stage such as the number of observations N or the size of

the hidden layer of the autoencoder H.
The small the parameters, the lower the number of parameters: learning the model and use it become very fast by reducing the dimensionality. Choosing N small means taking the risk to not have enough observations or have observations not sufficiently representative.



(a) Autoencoder architecture.

(b) Example of result after training the model autoencoder + Gaussian Process.

Fig. 3: In (3a) autoencoder neural network: $(p+1) \times R$ corresponds to the dimension of the input and the output layers and H_{AE} is the size of the hidden layer (parameters). In (3b), the plot to the top $\Pr(Y_{N+1} = 1|\mathcal{D}, \Theta, X_{N+1})$ in function of the time. The two other figures are two among five channels of the testing set. Green line is the ground-truth defined in the preprocessing step. The red line is the decision made by the GP classifier. The second channel is zoomed on a HVS.

For each rat, one signal session record has been used for the learning step and an other session has been used for the testing step. The criteria of performance for models are the sensitivity $Se = TP/(TP + FN)$ and the specificity $Sp = TN/(TN + FP)$, with TP is the number of true positive, TN is the number of true negative, FN is the number of false negative and FP is the number of false positive. The sensitivity gives the true positive rate: the number of correct detection under the number of correct detection and miss. The specificity give the true negative rate: the number of correct non detection under the number of correct non detection and false detection. We reproduce 5 times for each set of parameters the learning and the testing stages and compute the mean and the variance of Se and Sc to verify the performance and the robustness, regarding

random sampling.
Results are summarized in Tab.2. Data collection for each rat is different: the rat 2 provides data from channels M1U, M1D, SU and SD; rat 3 provides data from channels M1U, M1D, STRI and SD; rat 1 provides data from all channels[1]. Results are discuss in next section.

N	H	Rat number 1				Rat number 2				Rat number 3			
		Sensitivity		Specificity		Sensitivity		Specificity		Sensitivity		Specificity	
		mean	var	mean	var	mean	var	mean	var	mean	var	mean	var
50	10	.58	.0192	.88	.0161	.63	.0370	.47	.0596	.23	.0159	.83	.0082
50	30	.63	.0095	.96	.0031	.69	.0099	.57	.0212	.04	.0039	.97	.0019
50	50	.62	.0058	.97	.0008	.68	.0672	.52	.1624	.12	.0053	.91	.0056
50	100	.62	.0050	.96	.0013	.46	.0384	.78	.0531	.10	.0168	.93	.0047
200	10	.82	.0021	.95	.0016	.61	.0082	.56	.0044	.24	.0195	.86	.0061
200	30	.81	.0016	.90	.0032	.70	.0088	.71	.0087	.25	.0274	.82	.0128
200	50	.81	.0006	.91	.0097	.71	.0016	.73	.0057	.13	.0257	.90	.0173
200	100	.60	.0126	.96	.0097	.61	.0367	.82	.0104	.01	.0001	.99	.0000
500	10	**.85**	**.0010**	**.98**	**.0001**	.67	.0003	.65	.0036	.14	.0242	.94	.0043
500	30	.82	.0019	.95	.0006	.70	.0044	.67	.0043	.25	.0083	.86	.0028
500	50	.83	.0012	.89	.0005	.70	.0026	.68	.0017	.21	.0220	.85	.0141
500	100	.74	.0226	.94	.0054	.76	.0029	.70	.0005	.85	.0032	.85	.0027

Table 2: Result of the experience. N is the number of observations used from the training set. H is the size of the hidden layer of the encoder. Variance equal to .0000 in the table mean the value is less than 10^{-4}. Bold numbers highlight most relevant results.

5 Conclusion

Tab.2 highlights some tendencies in the parameters. First, the number of observations is critical for fine sensitivity and specificity. Taking too little oservation can alter the overall knowledge of the system : in this case the variance is often more important for $N = 50$ than for bigger N. But too much data leads to big with covariance matrices too long to calculate. Increasing H (hidden layer number of neurons) increases the sensitivity but decreases the specificity: the model tends to detect HVS all the time. On the opposite for small H we compress a lot of data by taking the risk to loose information. A large hidden layer better preserves the information but (i) the problem becomes hard to optimize (too much parameters) (ii) learning the model become time-consuming.
Results for rat 2 and 3 are not as fine as the first rat. By looking closely step by step signals of the two rats it appear that the intensity of the noise is much more important than in signals of the first rat. Means over channels of signal-to-ratio of the three rats are respectively, 35 decibels, 14 decibels and 10 decibels.

[1] see Tab.1 as a reminder.

Moreover, in rat 2 and 3, appearance of signals differs according to the various channels: some HVS do not appear in all channel which make the preprocessing step not relevant for those two rats. This is why results of rat 2 and 3 are not reliable to conclude with a high confidence level about the robustness of the model.
In a future work, we will develop an approach based on *unsupervised learning* model because by defining ourselves the groundtruth we may have missed some complex features in the signal which could have helped for predicting HVS. *Restricted Boltzmann* Machines is a promising stochastic model which, by exploring latent variables could find such hidden features.

References

1. Cyril Dejean, Christian E Gross, Bernard Bioulac, and Thomas Boraud. Dynamic changes in the cortex-basal ganglia network after dopamine depletion in the rat. *Journal of neurophysiology*, 100(1):385–396, 2008.
2. ERl Dorsey, R Constantinescu, JP Thompson, KM Biglan, RG Holloway, K Kieburtz, FJ Marshall, BM Ravina, G Schifitto, A Siderowf, et al. Projected number of people with parkinson disease in the most populous nations, 2005 through 2030. *Neurology*, 68(5):384–386, 2007.
3. Malte Kuss and Carl Edward Rasmussen. Assessing approximate inference for binary gaussian process classification. *Journal of machine learning research*, 6(Oct):1679–1704, 2005.
4. Martin Fodslette Møller. A scaled conjugate gradient algorithm for fast supervised learning. *Neural networks*, 6(4):525–533, 1993.
5. Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.
6. Carl Edward Rasmussen. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pages 63–71. Springer, 2004.
7. Vincent Vigneron, Tahir Qasim Syed, and Hsin Chen. Automatic detection of high-voltage spindles for parkinson's disease. In *BIOSIGNALS*, pages 372–378, 2015.
8. Jerrold L Vitek. Mechanisms of deep brain stimulation: excitation or inhibition. *Movement disorders*, 17(S3), 2002.